<ReactCheatsheet />

Hello World Component

```
class HelloWorld extends React.Component { // ES6 class that extends React's component class
  render() { // React uses the render method for deciding what the component should display
    return <h1>Hello World</h1> // JSX elements are similar to HTML tags
  }
}
// We attach the virtual DOM that React creates to the actual dom (in this case the element with the id root
ReactDOM.render(<HelloWorld />, document.getElementById("root"))
```

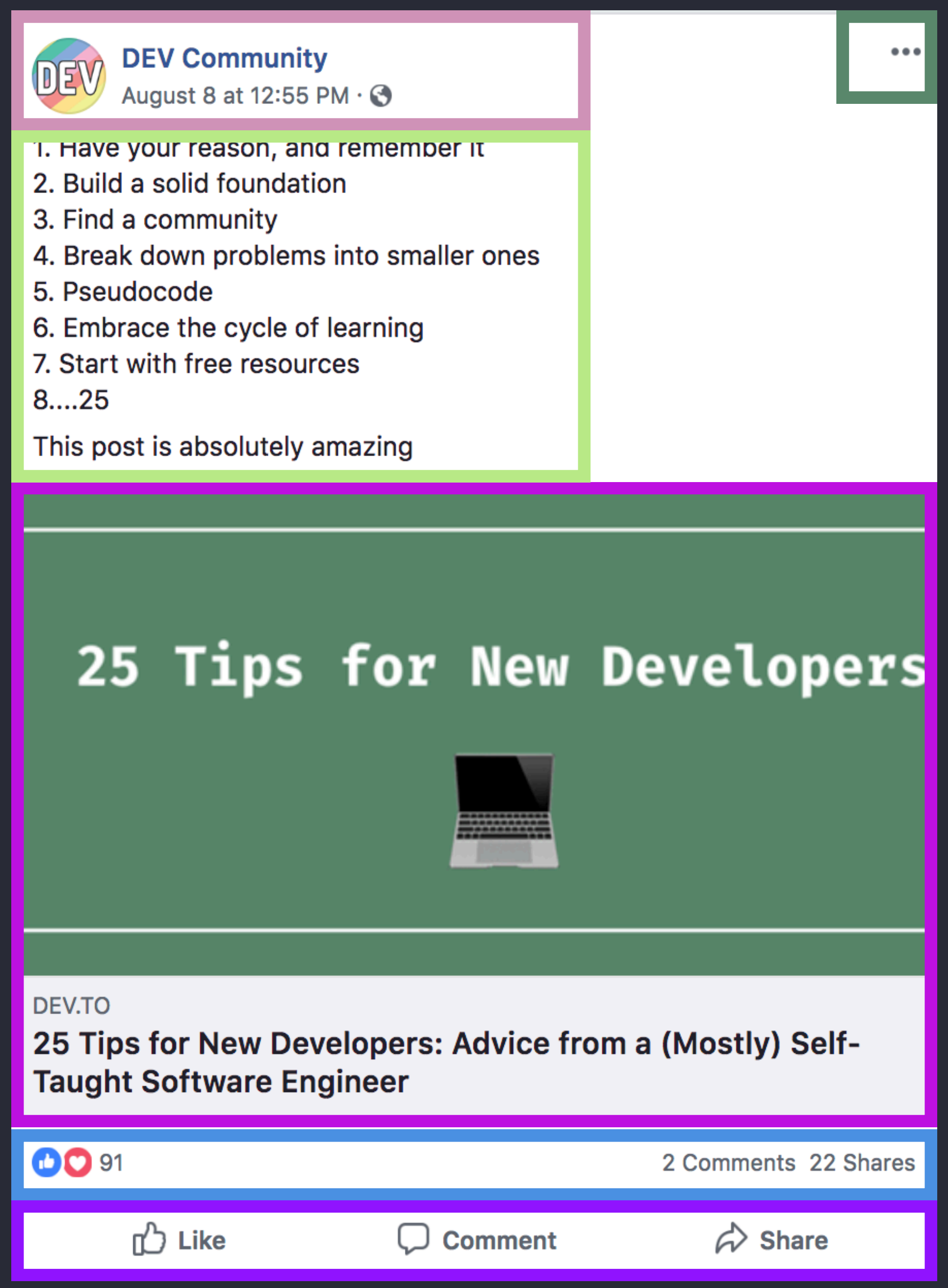
State and Props

In React, data is normally stored in two different ways: state and props. We use state for pieces of data that will change within a component. We use props to pass data from one component to another.

setState

When we use state, we never want to directly modify it. We instead use the setState method.

Component Architecture



DEV Community
August 8 at 12:55 PM ·

1. Have your reason, and remember it
2. Build a solid foundation
3. Find a community
4. Break down problems into smaller ones
5. Pseudocode
6. Embrace the cycle of learning
7. Start with free resources
8....25

This post is absolutely amazing

25 Tips for New Developers

DEV.TO
25 Tips for New Developers: Advice from a (Mostly) Self-Taught Software Engineer

91 · 2 Comments 22 Shares

In React, we structure our interface using components. These are reusable parts of the UI. Inside of those components, we also may have subcomponents, or components inside of components. We can use JSX elements to include a component within another one!

Stateless Components

If a component doesn't have state, we can make it a function-based component instead of using an ES6 class! Just return the JSX from the function instead of adding a render method.

Event Listeners

We can listen for events on JSX elements. One common catch with using event listeners in ES6 classes is that we need to preserve the context of the es6 class within the event handler. We can do this by binding `this` to the method in the constructor!

Some common events:

onClick

onChange

onScroll

onTouchStart

onKeyDown

Full Example

```
class Comment extends React.Component {
  constructor () {
    super()
    this.handleChange = this.handleChange.bind(this)
    this.state = {
      characterCount: 0
    }
  }

  handleChange (event) {
    this.setState({
      characterCount: event.target.value.length
    })
  }

  render() {
    return (
      <div>
        <textarea className="form-control" placeholder="Write a comment..." onChange={this.handleChange}/>
        <small>{this.props.maxLetters - this.state.characterCount} Remaining</small>
      </div>
    )
  }
}
```

Attributes with Different Names

Attributes work similarly in React as compared to HTML. Some names change though. First, any attributes with -'s in their names are converted to camelCase. Same with style values. Some other changes:

class => className

for => htmlFor

Passing Props

<Comment maxLetters={280} />